

Structuri repetitive în limbajul C++

Sunt utilizate în problemele a căror rezolvare necesită repetarea unor secvențe de un număr de ori, cunoscut de la început sau necunoscut.

În funcție de modul în care se verifică continuarea sau oprirea execuției structurile repetitive se clasifică astfel:

- a) cu număr cunoscut de pași sau cu contor
- b) cu număr necunoscut de pași
 - i) cu test inițial
 - ii) cu test final

- a) SR cu număr cunoscut de pași

Forma generală:

```
for(contor=vi;contor<=vf;contor++)  
    instrucțiuni;
```

sau

```
for(contor=vf;contor>=vi;contor- -)  
    instrucțiuni
```

Explicații:

- 1) **for** este cuvânt cheie pentru SR cu contor
- 2) contor este o variabilă a cărei valoare se modifică începând cu valoarea inițială și până la valoarea finală folosind formula **contor++** sau **contor-** - care sunt echivalente cu a scrie **contor=contor+1** sau **contor = contor - 1** adică la vechea valoare a contorului se adaugă 1 sau se scade 1.
- 3) în general variabila contor se notează cu litera **i**
- 4) instrucțiunile din interiorul structurii se execută de mai multe ori, numărul de repetiții se cunoaște de la început.

Exemple:

Exemplul 1

Exemplul 1 se poate rezolva cu structuri liniare astfel:

```
cout<<"Azi e miercuri."<<endl;  
cout<<"Azi e miercuri."<<endl;  
cout<<"Azi e miercuri."<<endl;  
cout<<"Azi e miercuri."<<endl;  
cout<<"Azi e miercuri."<<endl;
```

Concluzie: Acest mod de lucru este ineficient și consumator de resurse.

Exemplul 1 rezolvat cu structuri repetitive cu număr cunoscut de pași (cu contor):

```
for(i=1; i<=n; i++)  
    cout<<"Azi e miercuri."<<endl;
```

Pentru $n = 5$ se va afișa:

Azi e miercuri.

Azi e miercuri.

Azi e miercuri.

Azi e miercuri.

Azi e miercuri.

n	i	$i \leq 5$	cout $\Leftrightarrow$ (Afișare mesaj)	$i++$ ($i=i+1$)
5	1	A	Azi e miercuri.	2
	2	A	Azi e miercuri.	3
	3	A	Azi e miercuri.	4
	4	A	Azi e miercuri.	5
	5	A	Azi e miercuri.	6
		F		

Exemplele 2 și 3

2) for($i=1; i \leq n; i++$)

cout<< i <<" ";

Dacă citim pentru n valoarea 10 se vor afișa valorile: 1 2 3 4 5 6 7 8 9 10

3) for ($i=n; i \geq 1; i--$)

cout<< i <<" ";

Dacă citim pentru n valoarea 10 se vor afișa valorile: 10 9 8 7 6 5 4 3 2 1

Explicații:

Exemplul 2:

n	i	$i \leq n$	cout<< i	$i++$ ($i=i+1$)
10	1	A	1	2
	2	A	2	3
	3	A	3	4
	4	A	4	5
	5	A	5	6
	6	A	6	7

	7	A	7	8
	8	A	8	9
	9	A	9	10
	10	A	10	11
	11	F		

Exemplul 3:

n	i	i>=1	cout<<i	i- - (i=i-1)
10	10	A	10	9
		A	9	8
		A	8	7
		A	7	6
		A	6	5
		A	5	4
		A	4	3
		A	3	2
		A	2	1
		A	1	0
		F		

Exemplul 4

Afişaţi cinci * folosind structura repetitivă cu contor. (pe un singur rând şi apoi pe rânduri diferite)

```
for (i=1;i<=5;i++)
```

```
cout<<"*";
```

sau

```
for (i=1;i<=5;i++)
```

```
cout<<"*"<<endl;
```

n	i	i<=5	cout<<"*"	i++
	1	A	*	2

	2	A	*	3
	3	A	*	4
	4	A	*	5
	5	A	*	6
	6	F		
În final vom avea pe ecran:		*****		

Ce se va afișa pentru n=8?

```
1. cin>>n;
   s=0; x=2;
   for(i=1;i<=n;i++)
   {
       s=s+x;
       x=x+2;
   }
   cout<<s;
```

n	s	x	i	i<=n	s=s+x	x=x+2	i++	cout<<s
8	0	2	1	A			2	

2. Ce se va afișa dacă a=12 și n=10?

```
cin>>a>>n;
for(i=1;i<=n;i++)
    a=a+i;
cout<<a;
```

a	n	i	i<=n	a=a+i	i++	cout<<a
12	10	1	A		2	

Rezolvați folosind structuri repetitive cu contor:

- 1) Afișați pe ecran versul “- Ce te legeni, codrule,” - de 5 ori, pe rânduri diferite.
- 2) Afișați pe ecran &&&&&&&& de 10 ori.
- 3) Afișați numerele pare din intervalul [1,n]. Indicație: folosim $i=i+2$ în loc de $i++$
- 4) Să se afișeze numerele cuprinse în intervalul [100, 599].
- 5) Să se afișeze numerele pare cuprinse în intervalul [10, 99].